REMARKS

Claims 1-20 are pending in the present application. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Alleged Obviousness Based on Kaashoek and Auslander

The Office Action rejects claims 1, 10 and 19 under 35 U.S.C. § 103(a) as being allegedly unpatentable over M. Frans Kaashoek "The Logical Disk: A New Approach to Improving File Systems" in view of Auslander et al. (U.S. Patent Number 5,129,088). This rejection is respectfully traversed.

As to claims 1, 10 and 19, the Office Action states:

> **As to claim 1,** Kaashoek teaches device (physical disk, sec: 3/ 3.1/ Fig.2), device information (information about each block, sec: 3/ 3.1/ Fig. 2), transforming (divides the disk into large, fixed size segment and update its block number map by storing new physical of the block, sec: 3/ 3.1/ Fig. 2), a logical disk (logical disk, sec. 2.).
>
> Kaashoek does not teach device volume management, a device driver for a device. However, Auslander teaches the logical volume manage (col 9, ln 60-67 to col 10, ln 10-30/ col 13, ln 25-60), device driver (col 14, ln 50-67).
>
> It would have been obvious to apply the teaching of Auslander to Kaashoek in order to provide an improved data processing method for installing files on a data processing system.
>
> **As to apparatus of claim 10,** see the rejection of claim 1.
>
> **As to a computer program product of claim 19,** see the rejection of claim 1.

Office Action dated December 17, 2003, page 2.

Claim 1, which is representative of the other rejected independent claims 10 and 19 with regard to similarly recited subject matter, reads as follows:

> 1. A method for providing device management in a logical volume management system, comprising:
>
> receiving device information from a <u>device driver</u> for a device; and
>
> <u>transforming the device into a logical disk based on the device information</u>. (emphasis added)

Neither *Kaashoek* nor *Auslander*, either alone or in combination, teach or suggest transforming a device into a logical disk based on device information received from a device driver for a device. Furthermore, it would not have been obvious to modify

Kaashoek, Auslander, or any alleged combination of these references, to include such a feature.

Kaashoek discloses a log-structured logical disk implementation. Figure 2 of *Kaashoek* shows the data structures in the log-structured logical disk implementation, which include large, fixed-sized segments of a disk containing data blocks and a segment summary, a list table, and a block-number map. The list table stores the number of the first logical block for each list of logical blocks. The block-number map stores the physical address of a logical block, the successor to the logical block in its list, the length of the logical block, and whether the logical block is compressed. The block-number map maintains the mapping between the logical block numbers and physical disk-addresses. Using these data structures, file systems address blocks by their logical addresses, which do not change, even if the logical disk changes their physical locations.

The Office Action admits that *Kaashoek* does not teach device volume management or a device driver for a device. Applicants agree that Kaashoek does not teach this feature. Applicants further agree that Auslander teaches a logical volume manager that allocates physical partitions in a logical volume. However, neither reference teaches or suggests transforming a device into a logical disk based on device information obtained from a device driver for the device.

The Office Action alleges that *Kaashoek* teaches a logical disk implementation that divides the disk into large, fixed-sized segments (i.e. physical partitions) and that, simply because *Auslander* teaches a logical volume manager, that somehow the specific features of claim 1 are taught by this combination of teachings. While *Kaashoek* may teach dividing a disk into physical partitions, *Kaashoek* does not teach or suggest that such division is performed based on device information received from a device driver for the device. To the contrary, the details of the partitioning performed in *Kaashoek* are not described. Nowhere in *Kaashoek* is there any communication with a device driver of a device that is being transformed into a logical disk, such that the transformation into the logical disk makes use of information obtained from the device driver.

Furthermore, *Auslander* does not teach or suggest this feature either. *Auslander* teaches a mechanism for the creation of a logical volume which includes only the minimum number of physical partitions on the disk required to store the file system. As more storage space is needed by the file system, a logical volume manager allocates an additional physical partition to the logical volume. The individual physical partitions of the logical volume may be on different disk drives. As with *Kaashoek*, there is nothing in *Auslander* that teaches to obtain device information from a device driver for a device that is being transformed into a logical disk, and then performing the transformation of the device into a logical disk using the device information obtained from the device driver.

In the rejection of claim 1, the Office Action refers to the following portion of *Auslander* with respect to teaching a device driver:

> Another pair of special files designated /dev/hdn and /dev/pdn support the logical volume device driver and the physical device driver.
>
> The logical volume device driver and the physical volume device drivers provide block and character (raw) access to logical and physical volumes. The config device driver associates the minor device number to the logical volume. Normally, the special files, /dev/hdn and /dev/rhdn, are given the minor device number n. The logical volume with minor device number 0 is always the logical volume used to initially load the system program.
>
> When performing raw I/O, the number of bytes to be read or written should be a multiple of 512 bytes (a logical block). Likewise, 1seek system calls should specify a multiple of 512 bytes.
>
> All operations on a given DASD are performed at the driving site for (the volume group containing) that DASD. In a cluster configuration each site keeps a table mapping volume groups to driving sites.

Auslander, column 14, lines 50-67.

Auslander may disclose a logical volume device driver and a physical volume device driver, but *Auslander* does not teach or suggest transforming a device into a logical disk based on device information received from a device driver for the device. This portion of *Auslander* only teaches a logical volume device driver and a physical volume device driver provide access to logical volumes and physical volumes, respectively, and a config device driver associates a minor device number to a logical volume. Thus, *Auslander* teaches how to use a logical volume device driver, physical volume device driver, and config device driver to access logical volumes, but does not teach or suggest using such device drivers to obtain information that is used to transform a device into a logical disk. To the contrary, *Auslander* does not even teach or suggest a logical disk or transforming a device into a logical disk.

Thus, neither reference, whether taken alone in combination, teaches or suggests obtaining device information from a device driver for a device and then using this information to transform the device into a logical disk, as recited in claim 1. *Kaashoek* merely teaches that the log-structured logical disk implementation partitions a disk so that a partition may be maintained in main memory and written to in a single disk operation. *Auslander* merely teaches using device drivers to access logical volumes. Thus, any alleged combination of *Kaashoek* and *Auslander* still would not result in the features of claim 1 being taught or suggested.

Moreover, there is no teaching or suggestion in either of *Kaashoek* or *Auslander* regarding the desirability of combining these two systems in the manner alleged by the Office Action. As discussed in the previous Office Action, a "*logical disk*" and a "*logical volume*" are not the same entity and *Auslander* does not teach or suggest a "*logical disk*". *Auslander's* Logical Volume Manager provides the ability to create, modify and query *logical <u>volumes</u>*, physical <u>volumes</u>, and <u>volume</u> groups. Moreover, as stated in the Office Action, *Kaashoek* does not teach device volume management. The Office Action also states that *Kaashoek* does not teach constructing a volume group. Thus, it would not be obvious to one skilled in the art of how to combine the log-structured logical disk implementation of *Kaashoek* with the Logical Volume Manager of *Auslander*. The only teaching or suggestion to even attempt to combine *Kaashoek* and *Auslander* is completely based on a hindsight reconstruction having first had benefit of Applicants' claimed invention and disclosure.

Thus, neither *Kaashoek* nor *Auslander*, either alone or in combination, teach or suggest transforming a device into a logical disk based on device information received from a device driver for a device, as recited in claims 1, 10 and 19. In view of the above, Applicants respectfully request withdrawal of the rejection of claims 1, 10 and 19 under 35 U.S.C. § 103(a).

II. <u>35 U.S.C. § 103, Alleged Obviousness Based on Kaashoek, Auslander and IBM</u>

The Office Action rejects claims 2-9, 11-18 and 20 under 35 U.S.C. § 103(a) as being allegedly unpatentable over M. Frans Kaashoek in view of Auslander et al. (U.S.

Patent Number 5,129,088) and further in view of IBM (IBM to release LVM Technology to the Linux). This rejection is respectfully traversed.

Since claims 2-9, 11-18 and 20 depend from independent claims 1, 10 and 19, respectively, the same distinctions between *Kaashoek* and *Auslander*, and the invention recited in claims 1, 10 and 19, apply to dependent claims 2-9, 11-18 and 20. In addition, *IBM* does not provide for the deficiencies of *Kaashoek* and *Auslander* with regard to independent claim 1, 10 and 19. *IBM* is directed toward an email archive discussing Logical Volume Management technology for Linux. The *IBM* reference is cited for disclosing the AIX device manager, AIX partition manager and AIX feature plug-ins and for teaching accessing and using AIX logical volumes. The *IBM* reference does not teach or suggest the feature of transforming a device into a logical disk based on device information received from a device driver for a device. Thus, any alleged combination of the *IBM* reference with *Kaashoek* and *Auslander* still would not result in the invention recited in claims 1, 10 and 19 from which claim 2-9, 11-18 and 20 depend. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 2-9, 11-18 and 20 under 35 U.S.C. § 103(a).

In addition, *Kaashoek, Auslander*, and the *IBM reference*, either alone or in combination, do not teach or suggest the specific feature of transforming the logical disk into a logical partition, as recited in claims 3 and 12. The Office Action alleges that this feature is taught by *Auslander* at column 15, lines 52-68, which read:

> To map a logical address to a physical address, the logical volume device driver running at the driving site: (1) indexes into the logical volume map by logical volume number, (2) finds the first entry in the logical partition map corresponding to the first logical partition contained within the logical volume, then (3) looks up the physical volume and physical partition corresponding to the correct copy of the logical partition.
>
> The data structure described in FIG. 10 contains space for allocating additional logical volumes, and for allocating additional logical partitions to logical volumes. For example, entry 2 in the logical partition array has not been used, and may be used to allocated a third logical partition to logical volume 0.
>
> A device driver handling a physical request must map the physical address contained within the request into a device address before it can act on the request.

This portion of *Auslander* is directed to logical volumes, not logical disks as previously discussed. *Auslander* does not teach or suggest the extra layer of abstraction associated with transforming a physical device into a *logical disk*. *Auslander* teaches that a physical volume and physical partition corresponding to the correct copy of the logical partition is the location to look up a physical address. The present invention describes an additional layer referred to as a *logical disk*. *Auslander* does not teach or suggest this additional layer of a *logical disk* between a physical device and a logical partition. Therefore, *Auslander* does not teach or suggest transforming the *logical disk* into a logical partition. Further, the Office Action states that *Kaashoek* does not teach transforming the logical disk into a logical partition as recited in claims 3 and 12.

Thus, in addition to being dependent on their respective independent claims, claims 2-9, 11-18 and 20 are also distinguished over the *Kaashoek*, *Auslander* and the *IBM* references based on the specific features recited therein.

III. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: March 17, 2004

Stephen J Walder

Stephen J. Walder, Jr.
Reg. No. 41,534
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicants

SJW/vja